Design of a Phasor Data Concentrator for Wide Area Measurement System

Kedar V. Khandeparkar, Nitesh Pandit, A.M. Kulkarni
Dept. of Electrical Engineering, Indian Institute of Technology, Bombay
{kedark, anil}@ee.iitb.ac.in, panditnitesh@gmail.com

V.Z. Attar, S.U. Ghumbre
Dept. of Computer Science and Information Technology, College of Engineering, Pune
{vahida, shashi}@coep.ac.in

Abstract—**Wide area measurement system is playing a crucial role in improving the reliability of power system. Deployment of Phasor Measurement Units in Wide Area Measurement System has resulted in generation of a large amount of data at high rate across the PMU network. We propose a design and implementation of a Phasor data concentrator (named as iPDC) that can collect the data from different devices and direct it to real time applications, other iPDCs and also perform local archival for post processing.**

***Keywords*-Global Positioning System (GPS), Wide Area Measurement System (WAMS), Phasor Measurement Unit (PMU), Synchrophasor standard, iPDC, Simulator.**

I. Introduction

THE use of synchrophasors for monitoring and improving the stability of power transmission networks is gaining in significance all over the world. The aim is to monitor the system state, to increase the awareness of system stability and to make optimal use of existing lines. This way, system stability can be improved and overall transmission performance can be increased. The data from so many PMUs and PDCs needs to be collected and directed to proper channels for its efficient use. Thus, there is a need to develop an efficient data concentrator that can serve this purpose. Besides accepting the data from PMUs, PDC should be able to accept the data from other PDCs as well. We have designed such a PDC (iPDC) that accepts data from PMU and PDC that are IEEEC37.118 standard compliant.

The rest of the paper is organized as follows. Section 2 gives a brief introduction to WAMS components. Section 3 discusses the issues in WAMS. Section 4 describes the iPDC data and configuration frame format. Section 5 describes the iPDC design and implementation. Section 6 describes the PMU data storage in database. Section 7 describes the PMU simulator. Section 8 discusses the results. Section 9 concludes the paper and outlines our future research work.

II. COMPONENTS OF WAMS

Phasor Measurement Unit

The synchrophasors are recorded by the sensor devices called Phasor Measurement Units (PMUs). Phasors are time stamped using the GPS, and synchrophasors are gathered from a widely distributed transmission network at a central point, in order to suitably evaluate them there, as a monitoring instrument. The PMUs send the calculated phasor values by means of the standardized protocol, IEEE C37.118.

Phasor Data Concentrator(PDC)

The job of PDC is to collect the data from PMUs and or PDCs, time correlate them and feed them as a single stream to other application [1]. It provides additional functions as well. It performs various quality checks on the phasor data and inserts appropriate flags into the correlated data stream. It checks disturbance flags and records files of data for analysis. It also monitors the overall measurement system and provides a display and record of performance. It can provide a number of specialized outputs, such as a direct interface to a SCADA or EMS system.

III. Issues in WAMS

Delay

A PDC receives data streams from PMUs and other PDCs and correlates it in real-time into a single data stream that is transmitted to a computer via an Ethernet port. The propagation delays associated with communication links from a PMU to a PDC depends on the medium and the physical distance separating these components. In addition, there is a fixed delay associated with processing, concentrating, multiplexing, and transducers, and is independent of the communication [2].

Wait Time

PDCs also have a maximum wait time, typically of 1-4 seconds, to allow for all the PMU data to come in before aggregated data is outputted by the PDC. If the data from all the PMUs reach the PDC within this wait-time, it outputs the aggregated data right away. However, in the extreme case that the data from one of the PMUs is indefinitely delayed, and then the PDC will wait up to its pre-defined wait-time (i.e. 1-4 seconds) before the data is outputted by the PDC. Hence, the PDC can also introduce an additional delay equal to its wait-time if one of the PMU channel stop transmitting data to the PDC. Various details induced at PMU side are given in [3].

Vulnerability in WAMS Security

WAMS systems today utilize several common communication infrastructures - analog microwave, synchronous optical network (SONET), or virtual private network (VPN). Each of these communication methods contains vulnerabilities that can be used to interrupt communication or otherwise compromise the WAMS. Like most of today's SCADA systems, WAMS operate in an environment of complete and implicit trust. Neither C37.118 nor IEEE 1334 supports an authentication method. Without an authentication mechanism, a WAMS could be influenced by injected traffic being accepted and enacted by the PDC, state estimator, or other application [4].

IV. Data Format of iPDC

The latest PMU/PDC protocol is the IEEE C37.118 [5] that was developed in the last few years and approved late 2005. It will replace the IEEE 1344 synchrophasor protocol which has been in use as the PMU standard since its development in 1998. Before these standards were developed, the defacto standard for PMU to PDC communication has been the Macrodyne type 1 and type 2 protocols developed by Macrodyne Corporation. Some of the PDC to PDC protocols include the PDC data exchange format, the PDC stream, second level PDC using NTP time and the PDC stream, second level PDC using native time. These standards address issues like synchronization of data sampling, data to phasor conversions, and formats for timing input and phasor data output. As specified in IEEE C37.118 Synchrophasor Standard, there are four frame types, command, data, configuration and header. All these frames have the following fields in common- SYNC, FRAMESIZE, IDCODE, SOC, FRACSEC and CHK.

The PDC which has been designed (iPDC) follows the same standard [5] while combining the frames it receives. iPDC can receive(data and configuration) frames either from a PMU or a PDC. On receiving the frames, iPDC time-aligns them into a single frame. Those frames having same SOC and FRACSEC are combined into a single frame in the order of IDCODEs of PMUs/PDCs. A list of IDCODEs of PMUs/iPDCs is maintained. The IDCODEs in the list are in the order in which the iPDC receives configuration frames from PMU or other iPDCs. Thus the data frames that arrive out of order need to be aligned in the order of IDCODEs maintained in a list. These combined frames are then transmitted on a request from other iPDC. If iPDC receives data from other iPDC, then internal data of that frame is used for combining with main frame. This data would contain multiple PMU data. “Fig. 1” and “Fig. 2” show the combined configuration and data frames respectively.

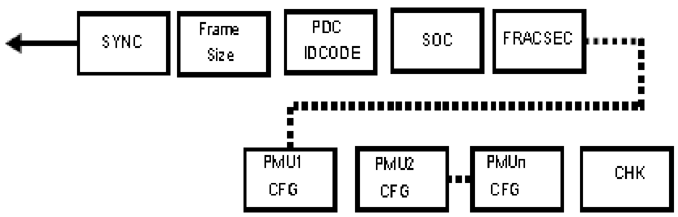

Figure 1. Combined Configuration Frame

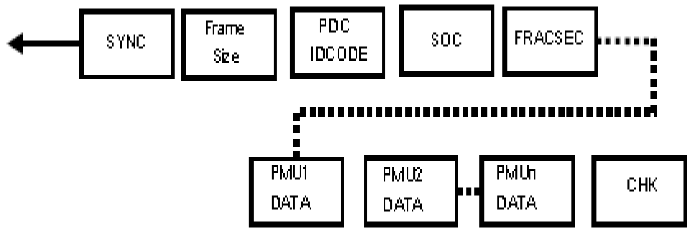

Figure2. Combined Data Frame

IPDC

WAMS Architecture

“Fig. 3” shows the WAMS architecture with iPDC and PMU at different levels. This architecture enables iPDC to receive data either from a PMU or other iPDC. Both PMU and iPDC from which the data is being received should be IEEE C37.118 synchrophasor standard compliant. As shown in the “Fig. 3”, iPDC at level 0 (substation PDC) receives data from PMUs and send the combined data to iPDCs at level 1 (central PDC). Level 1 iPDC accepts the data from iPDCs at level 0 and so on.

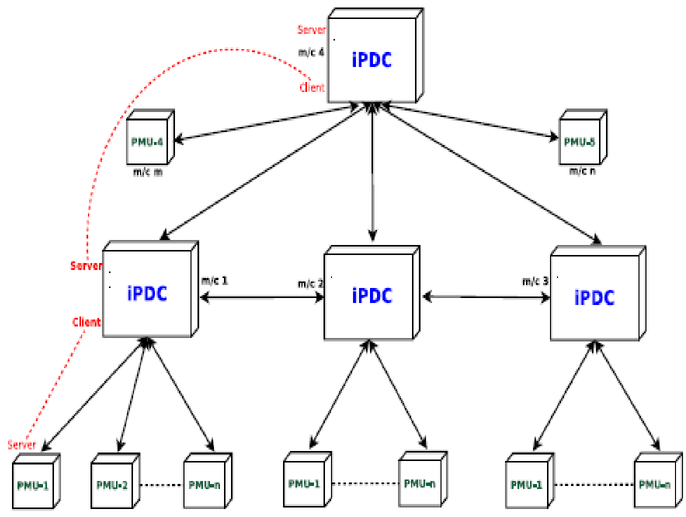

Figure 3. WAMS Architecture

iPDC Design

The client server architecture is common in networks when two peers are communicating with each other. Of the two peers (PMU and iPDC) that are communicating with each other in WAMS, one acts as a client and the other as a server. Since PMU serves the requests coming from iPDC by sending data or configuration frames, it is a server. It listens for command frames from iPDC. PMU-iPDC communication can be either over TCP or UDP communication protocols. On receiving command frames, PMU replies to the iPDC with data or configuration frames according to the type of request.

iPDC functionality is bifurcated as server and client. iPDC as a Client - When iPDC receives data or configuration frames its acts as a client. When acting as a client, it creates a new thread for each PMU or a PDC from which it is going to receive data/configuration frames. This thread would establish connection between the two communicating entities. It handles both TCP and UDP connections. The first frame that the server (PMU/iPDC) would receive is the command for sending the configuration frame. When the server replies with the configuration frame, iPDC (client) would generate another request to start sending the data frames. On receiving such a command frame, the server starts sending the data frames. If there is some change in the status bits of data frame which the client (iPDC) notices, it would take an action. For example if it notices a bit 10 has been set, it would internally send a command to server to send the latest configuration frame.

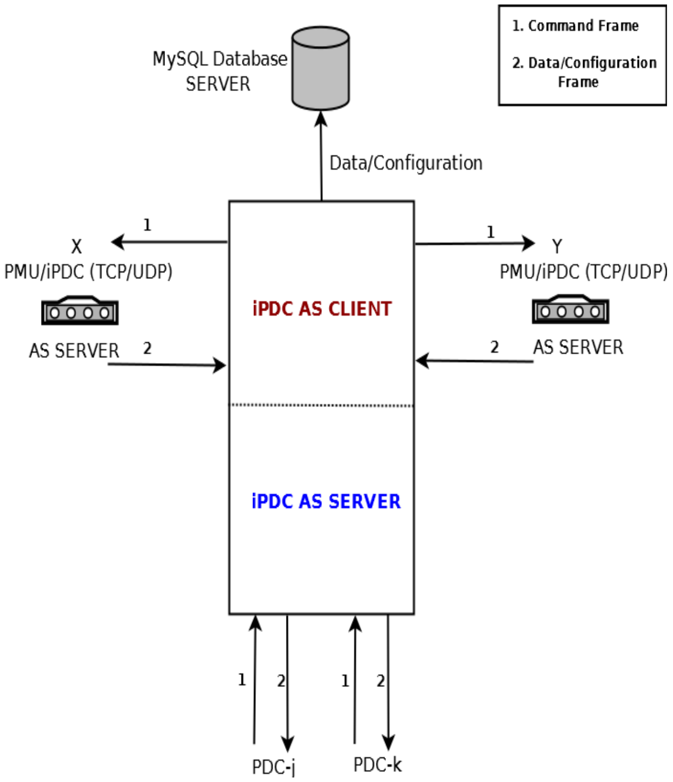

Figure 4. iPDC Design

iPDC as a Server- When iPDC receives command frames from another PDC it would acts as a server. There would be two reserved ports, one for UDP and another for TCP on which the iPDC would receive command frame requests. Thus PDC now plays the role of PMU waiting for command frames.

Time Alignment and Sorting

A circular FIFO array of fixed number of time-stamp buffers is maintained. Each received data frame with a time-stamp not present in the array is allocated a buffer from this array. All the subsequent data frames having the same time-stamp as arrived frame are then linked to the array index of the buffer. In this way a linked list of all data frames having the same time-stamp is maintained. If all buffers are full and a data frame with a new time-stamp arrives then as per the FIFO policy, the first arrive data frame need to be dispatched to make a space for newly arrived data frame.

Before dispatch operation, the frames need to be sorted so that they are sent in the proper order to the destination. Selection sort is used to sort the data frames as per their IDCODEs. This order of the IDCODE is the same as the order in which the first configuration frame arrives from each PMU/iPDC. After sorting, a combined frame is formed from the all the frames having the same time-stamp. Combined frame is then dispatched to other iPDC or a real time application. “Fig. 4” shows the details.

Security in WAMS

Security in WAMS is very important. As a minimal security, connection tables have been maintained at iPDC that authenticate each incoming packet. More work need to be done in this area to ensure cyber security. Data integrity is assured by performing checksum calculation of the received packets and verifying it with the checksum in the packets.

V. DATABASE DESIGN

MySQL and PostgreSQL Comparison

Among the widely popular open source databases MySQL and PostgreSQL, MySQL database is chosen for PMU data storage. PostgreSQL is a unified database server with a single storage engine. MySQL has two layers, an upper SQL layer and a set of storage engines. The most commonly used storage engines in MySQL are InnoDB. They provide full ACID support and high performance on large workloads. Applications can combine multiple storage engines as required to exploit the advantages of each. In PostgreSQL there is no built-in mechanism for limiting database size. This is the main reason why most of the web hosting companies use MySQL. More details and comparison can be obtained from [6].

iPDC Database

When iPDC receives data/configuration frames, it would direct the frames to a database server. The server may be on the local or remote machine. The database server process would have a IEEEC37.118 parser to parse configuration and data frames and create objects in memory. After parsing the data it would make entries in the configuration and data tables of the iPDC MySQL database. If a configuration frame comes for a newly added PMU it would be inserted in the configuration tables. If configuration frame for a previously added PMU arrives, then the previous entry in the tables is updated. The data frames are inserted as they come. This data which is stored in the tables can then be used for later analysis. The data from the database is archived periodically. This can be used for post analysis. “Fig. 5” shows the data storage process.

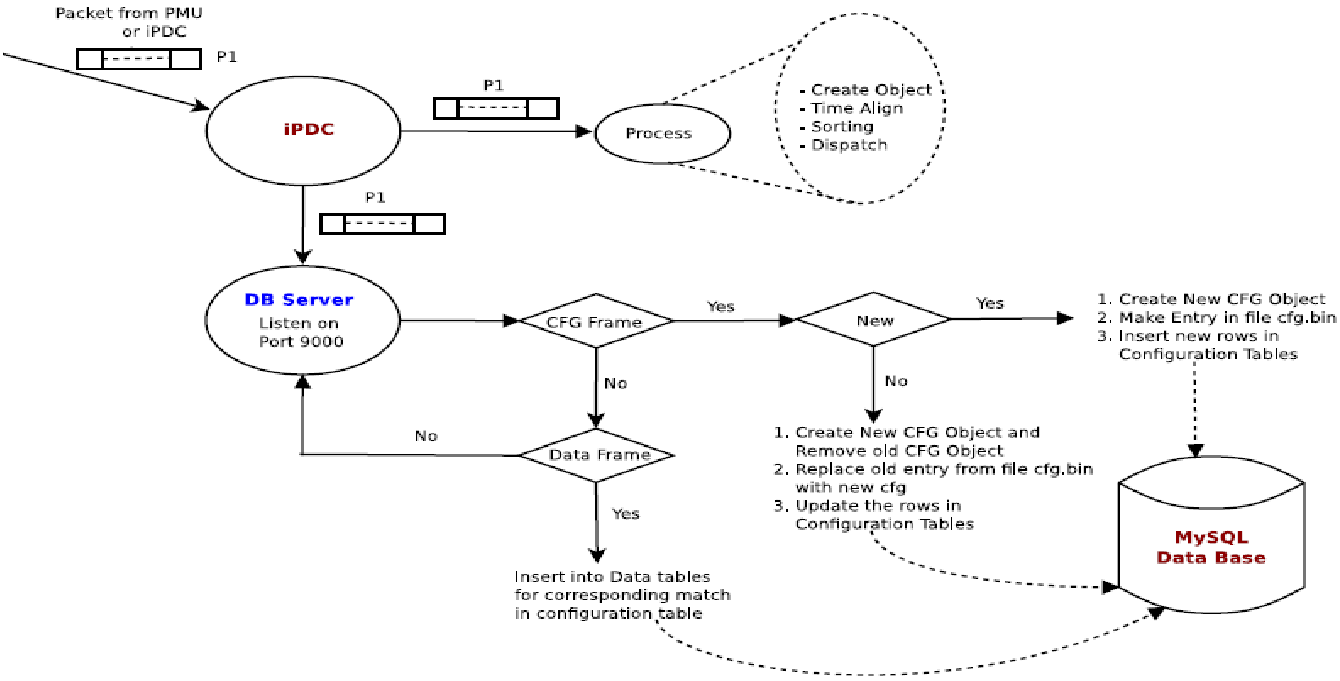

Figure 5. PMU data storage

VI. PMU SIMULATOR

PMU Simulator has been designed and implemented to test iPDC application. It generates all the frames given in [5]. PMU Simulator has been successfully tested with the PMU Connection Tester [7] and other PMU devices. It can emulate multiple PMUs on a single machine with different ports for UDP and TCP. The user can configure different parameters like phasor voltage, frequency, analogs, digitals etc. The data rate can also be varied. We can simulate multiple PMUs at different data rates and variable number of parameters. User has been provided the option to select the data from CSV file and generate the data frames.

VII. RESULTS

A setup was established with iPDC installed at two layers in WAMS topology. At layer 1 we had two iPDCs, each receiving data from 10 simulated PMUs. The combined frames from each of these iPDCs were then sent to another iPDC installed at layer 2. This iPDC was made to receive data from a PMU simulator. iPDC at both layers functioned well. The data from layer 2 iPDC was sent to MySQL database server. The query results from the database tables were verified with the configurations of simulated PMUs.

VIII. CONCLUSION AND FUTURE WORK

iPDC design proposed in the paper will serve to be the basic building block in the design of any other PDC. In future, we plan to test the iPDC and PMU Simulator in Real time operating systems like RTLinux, RTai etc., to check the

performance of iPDC in real time. We also need to ensure that the system meets the hard real time guarantees. There is a scope for improvement in database design that would enable data storage from multiple devices.

ACKNOWLEDGMENT

The authors thank Prof. S.A. Soman, Mr. Ken Martin, Mr. Gopal Gajjar, Mr. Anupam Chomal, Mr. Kalyan Dasgupta for their timely suggestions and help.

REFERENCES

[1] Andrew Armenia, Joe H. Chow, “A Flexible Phasor Data Concentrator Design Leveraging Existing Software Technologies”, IEEE Transactions On Smart Grid, June 2010.

[2] Chenine, M., Nordstrom, L., “Investigation of Communication Delays and Data Incompleteness in Multi-PMU Wide Area Monitoring and Control Systems”, Electric Power and Energy Conversion Systems, 2009. EPECS '09, Nov-2009.

[3] Kenneth Martin, “IEEE Standard for Synchrophasor Data Transfer for Power Systems”, IEEE C37.118.2-2011 (Revision of C37.118-2005).

[4] M.D. Hadley, J.B. McBride, T.W. Edgar, L.R. O'Neil, J.D. Johnson, “Securing Wide Area Measurement Systems”, Pacific NorthWest National Laboratory, US Dept. of Energy, June 2007.

[5] Kenneth Martin, “IEEE Standard for Synchrophasors for Power Systems”, IEEE C37.118 -2005 (Revision of IEEE 1344-1995), 2006.

[6] MySQL, “http://www.wikivs.com/wiki/MySQL_vs_PostgreSQL”.

[7] PMU Connection Tester, http://pmuconnectiontester.codeplex.com/”.

[8] Moustafa Chenine, “Analysis of Data Quality Issues in Wide Area Monitoring and Control Systems”, Bulk Power System Dynamics and Control (iREP) - VIII (iREP), 2010 iREP Symposium, Aug 1-6, 2010.

[9] Yingchen Zhang, “Wide-Area Frequency Monitoring Network (FNET) Architecture and Applications”, IEEE Transactions on Smart Grid, Vol. 1, No.2, Sep 2010.

[10] iPDC, “http://ipdc.codeplex.com.